

# Programming Concepts for Artists and Designers (3 credits)

Autumn 2009, #10226-2

TR, 5:30 – 7:18pm

1224 Kinnear, Rm. 205

*instructor:* Peter Gerstmann (pgerstma@accad.ohio-state.edu)  
*website:* <http://www.accad.osu.edu/~pgerstma/class/pca/>  
*office hours:* TR, 7:18 – 8:00pm; by email  
*location:* 1224 Kinnear, Rm. 205

## Prerequisites

Graduate student standing or permission of instructor.

## Description

This course teaches programming concepts useful for algorithmic manipulation of computer graphics data and software.

## Learning Goals

Artists and designers have digital tools to assist them in almost every medium. Many of these tools are controllable or extendable through the use of programming languages to achieve more specific results and more efficient processes.

Via ActionScript3, students will be introduced to fundamental programming concepts as they relate to visual results and the tools and processes common to their workflow. Students will apply these concepts to create work in Adobe Flash that is relevant to their discipline.

## Methodology

This course will introduce students to fundamental concepts and strategies for programming that relate to achieving graphical results. Code examples will be given to illustrate various concepts and techniques, but students will learn primarily by writing and discussing their own code. The class format will follow a general schedule of lecture, demonstration, homework, critique, discussion. Examples will be presented in lectures and demonstrations. Students will present their homework in critique sessions, to facilitate discussion of the topic at hand.

Students must demonstrate satisfactory achievement of course objectives through fulfillment of course projects and by contributing to class discussions and critiques. Course projects will require students to use a wide variety of software and equipment to produce imagery and code. Collaboration between students in the course and other faculty, staff and students at ACCAD is encouraged.

## Required and Optional Materials

There are no textbooks required for this class, but a bibliography is provided for further investigation.

## Weekly Topics

### Week01

R: **program** – what is programming?

assigned: *task01(R)*

### Week02

T: **object** – what mental models guide our code?

R: **expression** – how do we detail complex instructions?

due: *task01(T)*

assigned: *task02(T)*

due: *task02(R)*

### Week03

T: **instance** – how do we use pre-written code?

R: **repetition** – how do we compress long lists of similar instructions?

assigned: *task03(T)*

### Week04

T: **choice** – how can we execute some code only under certain conditions?

R: **composition** – how do we assemble imagery?

due: *task03(T)*

assigned: *task04(T)*

### Week05

T: **class** – how can we define our own re-usable code?

R: **encapsulation** – how do we hide minutiae to keep the code manageable?

due: *task04(R)*

assigned: *task05(R)*

### Week06

T: **event** – how do we react to the user?

R: **gesture** – what signals does the mouse generate?

due: *task05(R)*

assigned: *task06(R)*

### Week07

T: **time** – how can we observe time and react to its passing?

R: **text** – what signals does the keyboard generate? And how do we write messages to the user?

### Week08

T: **data** – how do we feed information to our applications?

R: **sound** – how can we make noise?

due: *task06(R)*

assigned: *task07(R)*

### Week09

T: **motion** – what are tweening and interpolation?

R: **wave** – why are waveforms useful for animation?

due: *task07(R)*

assigned: *task08(R)*

### Week10

T: **proposal** – what will you create for your final project?

R: **holiday** – Thanksgiving

due: *task08(T)*

assigned: *task09(T)*

### Week11

T: **work** – in-class work on final project

R: **work** – in-class work on final project

due: *progress report01(T)*

due: *progress report02(R)*

### Finals

T: optional work day

R: **final**

due: *Final Presentation (R)*

## Course Grading Scale

A	=	100%	-	95%
A-	=	94%	-	90%
B+	=	89%	-	87%
B	=	86%	-	84%
B-	=	83%	-	80%
C+	=	79%	-	77%
C	=	76%	-	74%
C-	=	73%	-	70%
D+	=	69%	-	64%
D	=	63%	-	60%
E	=	59%	-	0%

## Course Assignments

Students will be given an overall grade based on their coursework weighted as follows:

- 20% daily challenges
- 40% homework tasks
- 40% final project

## Grading Policy

Students must demonstrate satisfactory achievement of course objectives by skillfully completing course assignments and contributing to class discussions and critiques. Course assignments will require students to use a wide variety of software and equipment to produce video games. Collaboration between students in the course and other faculty, staff and students at ACCAD is encouraged.

Adherence to deadlines is expected. It is the individual student's responsibility to keep track of deadlines and to present the work to the class and instructor on the specified dates. Late assignments will have their grade lowered by one level per missed class (e.g. B to B- for an assignment one class late).

Students choosing to use "at home" hardware and software must have their current working files on the course system and available for review at the beginning of each and every class. Problems with home systems or incompatibilities will not be an acceptable excuse for missed goals. Technical problems will happen frequently during the quarter and students will have trouble accessing the computer lab during "prime time" hours. Students must make their own arrangements for overcoming these difficulties and submitting their work on time. Unless there is a complete system failure in a computer-related course, technical difficulties are never an acceptable excuse for not meeting a deadline. Students should plan their time and workload to anticipate the technical hurdles that are a part of this profession.

## Attendance Policy

All students are required to be on time and in attendance for each and every class. Students arriving to class more than 15 minutes late will be counted as absent. Two absences will lower a final grade by one level (e.g. B to B-), three absences will lower a final grade by one letter (e.g. B- to C-) and four absences will result in failure of the course.

## Academic Dishonesty

Any and all suspected cases of academic dishonesty will be dealt with according to university procedures. Students are referred to the student handbook for further information on academic dishonesty and the accompanying procedures and penalties.

Students can read the code of student conduct at:

<[http://studentaffairs.osu.edu/resource\\_csc.asp](http://studentaffairs.osu.edu/resource_csc.asp)>

## Personal Safety

The University Escort Service operates after 6pm until 3am when classes are in session (i.e. not during quarter breaks and University holidays), and will assist OSU students who live off campus as well as on campus. The University Escort Service can be contacted at 614-292-3322, and scheduled pick-ups are taken in advance.

## **Accommodations for Students with Disabilities**

**Any student who feels s/he may need an accommodation based on the impact of a disability should contact the instructor privately to discuss their specific needs. Please contact the Office for Disability Services at 614-292-3307 in room 150 Pomerene Hall to coordinate reasonable accommodations for students with documented disabilities.**

### **Course Bibliography**

Adobe. *ActionScript 3.0 Language and Components Reference*.

[<http://livedocs.adobe.com/flash/9.0/ActionScriptLangRefV3/>](http://livedocs.adobe.com/flash/9.0/ActionScriptLangRefV3/)

Online Documentation of the adobe, air, fl, and flash packages.

Adobe. *Flex 3.0 Language Reference*.

[<http://www.adobe.com/go/AS3LR/>](http://www.adobe.com/go/AS3LR/)

Online Documentation of the air, flash, and mx packages.

Colin Moock. *Essential ActionScript 3.0*. O'Reilly (June 2007).

ISBN 0-596-52694-6.

John Maeda; Paola Antonelli. *Design By Numbers*. The MIT Press, 2001.

ISBN 0-262-63244-6

Joey Lott; Darron Schall; Keith Peters. *ActionScript 3.0 Cookbook*. O'Reilly (October, 2006).

ISBN 0-596-52695-4.